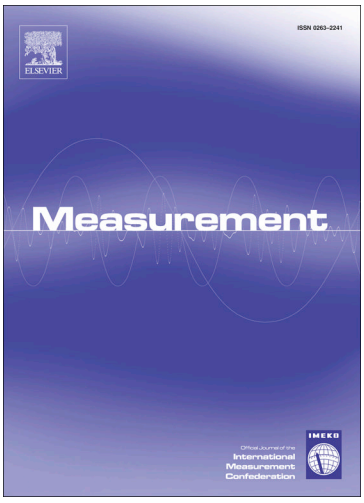


# Investigating the Quality Inspection Process of Offshore Wind Turbine Blades Using B-Spline Surfaces

Dominic Auger, Qing Wang, Jon Trevelyan, Songling Huang, Wei Zhao

PII: S0263-2241(17)30652-8
DOI: https://doi.org/10.1016/j.measurement.2017.10.027
Reference: MEASUR 5029

To appear in: *Measurement*

Received Date: 24 June 2017
Revised Date: 9 October 2017
Accepted Date: 11 October 2017

Please cite this article as: D. Auger, Q. Wang, J. Trevelyan, S. Huang, W. Zhao, Investigating the Quality Inspection Process of Offshore Wind Turbine Blades Using B-Spline Surfaces, *Measurement* (2017), doi: https://doi.org/10.1016/j.measurement.2017.10.027

# Investigating the Quality Inspection Process of Offshore Wind Turbine Blades Using B-Spline Surfaces

Dominic Auger[1], Qing Wang[1*], Jon Trevelyan[1], Songling Huang[2] , Wei Zhao[2]

[1]Department of Engineering, Durham University, DH1 3LE, U. K.

[2] Department of Electrical Engineering, Tsinghua University, Beijing, 100084, P.R. China.

***Abstract*—** Wind turbines can only extract their rated amount of power if their blades conform closely to the Computer Aided Geometric Design (CAGD). This is quantified by reconstructing B-spline surfaces from measurement data taken from the blade. For reliable comparisons between the surface and CAGD, the generated surface must be an accurate representation of the part. To do this, the input parameters to the fitting process need to be optimized. Previously this has proved to be time consuming and computer intensive. This paper has focused on presenting a protocol for control point locations that increases the initial surface fitness; therefore, decreasing the time taken for an optimiser to converge on the ideal locations. The presented protocol was found to increase the fitness of a surface by up to 150%. For low tolerance products it has been observed that the protocol could remove the need for an optimiser all together.

*Index Terms*— B-Spline, surface fitting, quality control, control points, optimiser.

## 1. INTRODUCTION

With limited non-renewable resources and a change in public perception, the UK government has announced that by 2020, 15% of energy will be provided by renewable sources [1]. As the UK is surrounded by ideal locations for offshore wind farms, a target of 18 GW of installed capacity has been set for 2020[2].

To meet these targets it is imperative that wind turbine (WT) blades not only conform closely to their Computer Aided Design (CAD), but are also fit to operate at any given time. This means reducing the turbine's annual downtime. As manufacturing defects are the most common

reason for blade failure [3], this is achieved by identifying them before the blades go into operation. In a further survey conducted over five wind farms it was found that 7% of all WT blades had been replaced [3]. With WT blades making up 22.2% of the total component cost [4], replacing them during operation is not only difficult but also expensive. This was emphasised by the Suzlon Blade recall which saw a total of 1,251 2.1 MW blades being removed for strengthening following the appearance of cracks on the surface [5]. It is estimated that the retrofit program cost Suzlon Energy Ltd. $25 million (USD) [5].

To mitigate blade failure, and ensure that the WT produces their rated amount of power, blades go through a quality control (QC) stage after production. Within this, areas that fall outside of the specification are identified along with defects. The ability to detect defects is the corner stone of QC, and works on the logic of comparing a fabricated component to its original CAD drawings. It is through variations between the two that distinguish surface defects and out of specification regions.

Finding these variations is the backbone of reverse engineering. This is conducted by taking measurement data, collected by a metrology device, and then fitting them to a surface. This can then be compared to the original CAD surface. However, after a surface has been generated, it needs to pass through an optimizer. This increases the likeness between the generated surface and its data points. It has been observed that surface optimization for large scale manufactured parts requires a high level of computer power and long computational times, resulting in long through-put times of the Quality Control stage. As a result, this paper focuses its research on finding an optimal set of original setup constraints for B-spline surface implementation, which may then reduce the through time of the later optimizer.

## 2. METROLOGY THEORY

### *2.1 Contact & non-Contact*

To quantify whether a WT blade is inside its tolerance, measurements must be taken from the blade. This is achieved using metrology devices. Fundamentally, metrology hardware can be broken down into two bodies: contact and non-contact equipment. Contact equipment requires physical contact with the measurement surface, normally achieved through a probe.

Commonly, non-contact machines emit structured light [6], which is reflected back from the part and recaptured by the unit. Selecting the right device is not easy, and depends largely on the part being measured.

Although measurement rates of contact devices are slow [7], their gathered data is generally more accurate than non-contact devices [8,9]. However, most contact devices require components to be taken to them [10], making them unsuitable for field inspection. Furthermore, a confined measurement volume along with a permanently fixed probe can cause difficulties during inspection of large parts [11], such as WT blades. To overcome the drawbacks of one device, multiple devices have been connected within the same system. This was found to reduce the risk of measurement errors [10].

Due to these factors, it is common practice to see non-contact devices being used for large parts such as WT components, ship hulls and many aerospace components. The measurement data collected by metrology devices makes it possible to reconstruct a graphical representation of the part through spline surface fitting. From this, comparisons to the original CAD drawings can be made.

*2.2 Splines*

To reduce the probability of blade failure and thus decrease the annual downtime of a WT, blades go through a quality inspection (QI) process. This ensures that a blade conforms to its CAGD within a given tolerance. Most modern WT blades are manufactured out of composite materials. This is commonly achieved through Vacuum-Assisted Resin Transfer and Pre-Impregnated resin [3].

By observing variations between a part and its original CAD model, the capability of the manufacturing tools and the materials can be verified. Each manufacturing machine will have limitations, leading to these variations. Materials also respond differently to conditions they are subjected to. In the composite layering process, porosity and in-plane and out-of-plane waviness can occur [12]. It was found that as little as 1% porosity can decrease the material's strength by 7% [13]. An effect such as porosity can cause blade failure during the operating life. QI therefore verifies the efficiency of the manufacturing process, whist validating the selection of materials and manufacturing tools. In addition, it increases the probability that the blade will

operate for the full duration of its expected life. As a result, thorough QC and QI decreases the chance of encountering unexpected cost during the product's operation life.

To ensure this, these variations must first be found. Finding these variations can be achieved through constructing a curve/surface from measurement data.

*2.3 Bézier curves*

The simplest method of generating a curve or surface is through the Bézier method. This paper refers to curves as 2D lines in the x-z plane, whilst the surface is a 3D representation in the x, y & z space. To implement a Bézier Curve, Equation 1 is used:

$$C(u) = \sum_{i=0}^{n} b_{i,n}(u) v_i \qquad 0 \leq u \leq 1 \qquad (1)$$

Where *C(u)* is the Bézier Curve, $b_{i,n}(u)$ is known as a Bernstein polynomial, $v_i$ are the control points (CP) and *u* the parametric location values. The subscripts *i* and *n* refer to the point of interest and the degree of the system respectively. The degree is the highest power to which a derivative is raised, whilst the order refers to the total number of coefficients equal to one plus the degree. In order for these curves to be implemented, *n*+1 control points are needed.

Although easy to implement, Bézier Curves have associated disadvantages making them unappealing for mapping complex free-form surfaces. Bézier Curves can only change direction one magnitude less than their order. For multiple changes in direction, the order of that curve needs to be large. This increases the computation time and complexity of implementation. Bézier Curves also lack the ability to be locally modified. Changes to the location of a CP will affect the global geometry of the curve or surface [14]. To overcome these shortfalls, multiple Bézier Curves can be joined together to create curves of higher complexity. These composite curves are referred to as B-Splines, or Basic-Splines.

B-spline functions are widely used in applications in Computer Aided Geometric Design, image processing, reverse engineering and computer graphics. To employ B-spline curves for wind turbine blades, a knot vector will need to be defined in advance. Based on this the control points are identified to minimise the least squared error between the data points and the function.

*2.4 B-Splines*

B-Splines were first introduced by Schoenberg [15] in the late 1940s, however they were rarely used until De Boor published his papers in the early 1970s [16]. A B-Spline curve is an accumulation of multiple Bézier Curves, otherwise known as polynomial pieces (pp). The governing equations for a B-spline Curve and Surface can be seen in Equations 2 and 3 respectively.

$$B(u) = \sum_{i=0}^{n} N_{i,p}(u) v_i \quad (2)$$

$$B(u,\mu) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_\mu} N_{i,k}(u) N_{j,q}(\mu) v_{i,j} \quad (3)$$

Where *B(u) and B(u,μ)* represents the parametric B-spline, $N_{i,p}(u)$ *and* $N_{j,q}(\mu)$ the normalised basis functions, *k and q* are the orders whilst $n_u$ *and* $n_\mu$ are the total number of control points (CP). Surface generations can be thought of as combining two splines normal to each other, in the parametric directions *u* and $\mu$.

The basic function can be treated as a triangular scheme. This is represented in the Cox-de Boor recursion equation shown in Equations 4 and 5.

$$N_{i,0}(u) = \begin{cases} 1 \ \ if \ t_i \leq u \ \leq \ t_{i+1} \\ 0 \qquad otherwise \end{cases}, \quad (4)$$

$$N_{i,p}(u) = \frac{u - t_i}{t_{i+p} - t_i} N_{i,p-1}(u) + \frac{t_{i+p+1} - u}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(u) \quad (5)$$

Where $t_i$ is a knot of knot vector *T*. Each CP is associated with a basic function. Combined together they form a curve or surface.

*2.5 Control points*

It has been found experimentally that the number, density and locations of the CPs has a direct effect on the fitness of the resultant curve. Despite this, most algorithms for B-splines implementation will arbitrarily pick the number, density and locations of the CPs. The fit of the surface will then later be increased by either introducing rationality or passing the CP location through an optimiser. With large numbers of CPs these methods have proven to be time consuming. As a result, the time spent in the QI stage has increased dramatically. In some cases, when tolerance limits are low, the amount of time taken to compute the optimal CP location cannot be justified.

It has also been observed that most fitting optimizers only work with the locations of the CP. However, during the current study it was found that the density and number of CP also had a dramatic effect on the likeness between the generated surface and the CAGD. Building on this, CP selections can be linked to computational time. Due to this, the current research aims to present a set of guidelines for CP location, density and number. Furthermore, for parts that have higher tolerances, selecting suitable CP the first time round will reduce the time taken for a later optimizer to converge at the optimal CP location. This then leads to a reduced through-put of the quality inspection process.

### *2.6 Knot vectors and values*

#### *2.6.1 Uniform knot vectors*

Once the number of CPs has been decided a knot vector can be created. A knot vector, or knot sequence, is a string of non-descending knot values. These create the joins between pp. Knot vectors, seen in Equation 6, can be broken down into two groups: uniform and non-uniform knots. Here, uniformity is reached if the span between any two knots is constant over the entire knot vector.

$$T=\{t_i,t_{(i+1)},\ldots,t_{(m-1)},t_m\} \quad (6)$$

Here m is the total number of knot values, which can be calculated using Equation 7.

$$m=n+k \quad (7)$$

Equation 7 relates the number of CPs and knots together to the order. To fully define a B-Spline

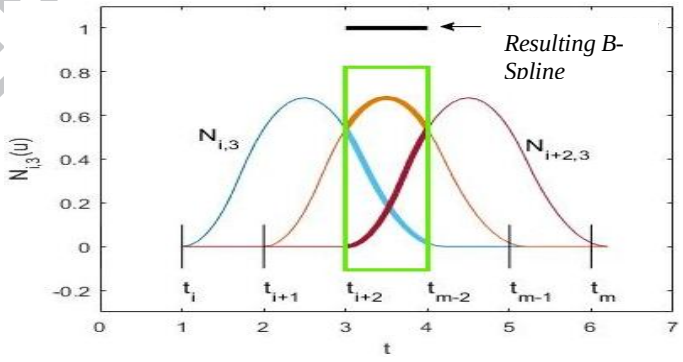


Figure 1. B-spline basis function with uniform knot

over some range, there must be at least three basic functions in that range.

Figure 1 shows three basic functions formulated from a uniform knot vector. The green box represents the range where the B-Spline is fully defined, between knots $t_{i+2}$ –$t_{m-2}$. The rest of the B-spline curve would be truncated.

Uniform knot sequences will result in a B-spline that does not pass through the first and last control points. By setting the first *k* knots to $t_k$ and the last k knots to $t_n$, an open uniform knot vector is created. Figure 2 shows the effects of using this type of knot vector.

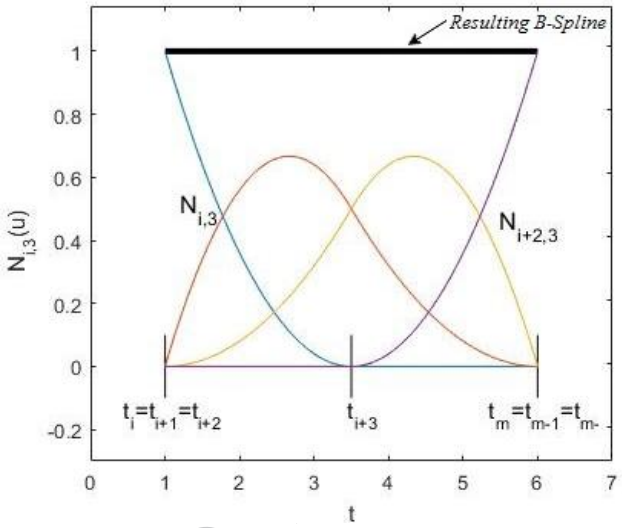


Figure 2. B-spline basis function with open uniform knot vector

The resulting B-spline is now defined over the entire vector, i.e. $t_i - t_m$.

As the complexity of a surface increases, the density of control points, and thus the knot, must also increase. However, whilst using uniform knot vectors the spacing must remain constant.

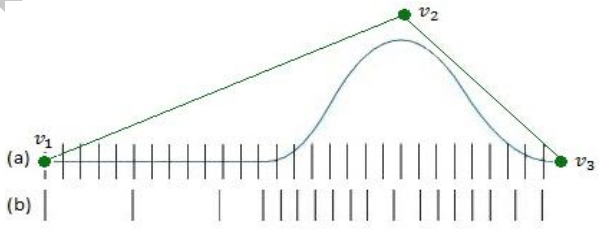


Figure 3. (a) Uniform knot vector used to map blue curve, (b) non-uniform knot vector

This often leads to too many knots, as seen in Figure 3 (a), resulting in long, expensive computations and over-fitting of the data [17]. Furthermore, the spread between measured points often has large variations. Using a uniform knot vector to interpolate this sort of data can lead to overshoot, oscillations and loops [18]. This can be overcome by using non-uniform

knots, which can be seen in Figure 3 (b). In Figure 3, the blue curve is defined by the knots (black vertical lines) and the three CP (green dots).

*2.6.2 Non-uniform knot vectors*

Non-uniform knot vectors are only constrained by a non-descending sequence. This allows the density of knots to be locally modified as a surface's complexity changes, as seen in Figure 3 (b). In other words, it allows a user to modify the curve locally without having an effect on the global curve. This then reduces both the cost and time of computation whilst increasing the fit of the B-spline. Non-uniform knot vectors are used extensively when dealing with measurement data.

*2.7 Parameterization*

All of the splines when first generated will need to be fitted to the set data points. Before an optimal curve/surface is created the splines will have to go though some fitting process. Normally, in the case of the B-spline this will take the form of a Least Square Fitting method (LSQ), which requires parameter values in order to work. As in most cases of reverse engineering the data points would have been collected by some metrology device, this means that the gathered points will be digitized, and therefore, cannot go straight into a LSQ algorithm. Before this is possible the points need to be first parameterized, which can be achieved through a number of different parameterization methods. The most common however are: cumulative chord length, centripetal and uniform parameterization methods, which will be titled 'the classical models'. In 1989, Lee [19] produced a formula that encompassed all three classical methods, seen in Equation 8.

$$u_1 = 0,$$
$$u_i = u_{i-1} + \frac{\|P_i - P_{i-1}\|^e}{\sum_{j=1}^{N-1}\|P_j - P_{j+1}\|^e}, \quad 2 \leq i \leq N \qquad (8)$$

Where P represents the digitized measured points, $u_i$ are the parameter location values, N the total number of measurement points, and $e$ a measure of blending. Varying $e$ causes the equation to change between the three classical models. When e equals 0, 0.5 and 1, the model turns to uniform, centripetal and cumulative chord length, respectively. Associated advantages and disadvantages for each are given in Table 1.

| *Parameterization Method* | *Advantages* | *Disadvantages* |
|---|---|---|
| Cumulative chord length | Takes into account the relative spacing between points. | Twists away from the control polygon may be occur. |
| Centripetal | Can generate very smooth curves/surfaces. | Small loops may be produced. |
| Uniform | Easy to implement. | Doesn't take spacing into account. |

Table 1. Comparison between classical parameterization methods

Using uniform parameterization, the values of $u_i$ become equally spaced. This method does not take into account the relative spacing of the measurement points. The chord length method however does, and thus was found to produce better results in [20]. It is also more capable of avoiding self-intersections [21]. If the data points are roughly equally spaced then the chord length method can be approximated by the uniform method [22]. The centripetal method is a mixture of the other two and works by observing the changes in the curvature for curve fitting [22].

No one parameterization method can claim to generate maximum fitness of every data set. As a result, before any surface generation was undertaken, the optimal amount of blending for each data set had to be found.

As table 1 shows there are more than just the classical models available to use. Ma and Kruth's [22] novel approach in 1995, of projecting points from an underlying curve or surface along some vector to a base curve or surface, saw both stable and improved results over other models for both regular and irregular distributed points. The basic principles behind the method for a surface are as follows. Firstly a base surface is constructed. This can be any parametric surface, but must follow two main conditions. Number one is the unique local mapping property, local mapping means that there exists a closed curve on which all the projected points lie and the uniqueness infers that one point on the underlying surface can only have one point on the base surface. The second condition is smoothness and closeness. The base surface is required to be as simple and smooth as possible, whilst still resembling the

underlying surface. Often, if the unique local mapping condition is fulfilled then the closeness is also satisfied. Once this base surface is constructed, and it meets the two required conditions, then the measured digitized points can be projected onto it. This projection can be done either normally or along some vector to the base surface. These projected points can now be treated as the location parameters of the measured points. Once these are reordered on the base surface they can go through a minimisation algorithm to make the fit between the underlying surface and the base surface as close as possible.

*2.8 Fit of generated B-Spline surfaces*

Different methods can be found in literature on how to generate B-spline curves. Dung & Tjahjowidodo [23] presents a new strategy for optimal knot calculation in a B-spline fitting based on a local algorithm that is capable for non-uniform knot cases. Park [24] uses a B-spline to fit rectangular grid points, which is based on adaptive knot placement using dominant columns along u- and v- directions. Zhang et al [25] presented an efficient descent-based optimization algorithm for optimise knot position. A fast B-spline curve fitting has been introduced in [26] by employing a L-BFGS optimization method. Wang, Pottmann & Liu present a squared distance minimization method for computing a planar B-spline to approximate a target shape defined by point cloud [27]. Bo, Luo & Wang [28] use a graph-based method for reconstructing curves from a planar point cloud. The weighted graph representation of point clouds has been introduced to evaluate the segmentation results of data points by a shape evaluation function. The method we propose is to optimise the control point locations to increase the initial surface fitness.

Once a B-spline curve or surface has been constructed, its fit to the measurement data needs to be quantified. This was achieved through a fitness value. To calculate the fitness the Root Mean Squared (RMS) Error must first be calculated. This was achieved by calculating the residual error, $e_i$, for a number of locations along the fitted B-spline. Once obtained Equation 9 was used to produce the RMS error.

$$e_{RMS} = \sqrt{\sum_{i=1}^{S} \frac{{e_i}^2}{S}} \qquad (9)$$

*S* is the total number of residuals. From this it is possible to calculate the fitness value for the generated surface, seen in Equation 10.

$$Fitness = \frac{1}{1+ e_{RMS}} \quad (10)$$

Once a value is established the surface can go through an optimiser, with the aim to increase the fitness. This can be achieved using the Least Squares method (LSQ). For a B-spline curve, this is shown in Equation 11.

$$min\boldsymbol{E} \sum_{j=1}^{N} [P_j - \sum_{i=1}^{n} N_{i,n}(u_i) v_i]^2 \quad (11)$$

Whilst fixing the parameters $u_i$, it is possible to find the optimal set of control points $v_i$. However, using this proved to be time consuming. For the research scope, the surfaces and curves presented in the report have mostly been optimised manually. This was so the effects of editing any one input could be found, and easily linked to that single change in the inputs.

## 3. Measurement Method

### *3.1 Equipment and data*

Data used in this study were collected by Nikon's MV200 coherent laser radar (CLR) system, in [29, 30]. These data are the measurement points along the B-spline as shown in Figure 4. The measurement data represents two different WT blade models; the first, a simple D-shape section of a blade located 30m from the root. The second, a half sized section of Vestas's 44m long offshore WT blade, located 5.5m from the root. The first model offers data relating to one surface, while the second provides data from the front, base and back faces of the blade. A comparison between the two models can be seen in Figure 4. For both models, measurement data was collected along multiple vertical lines, making it ideal for B-spline visualisation.

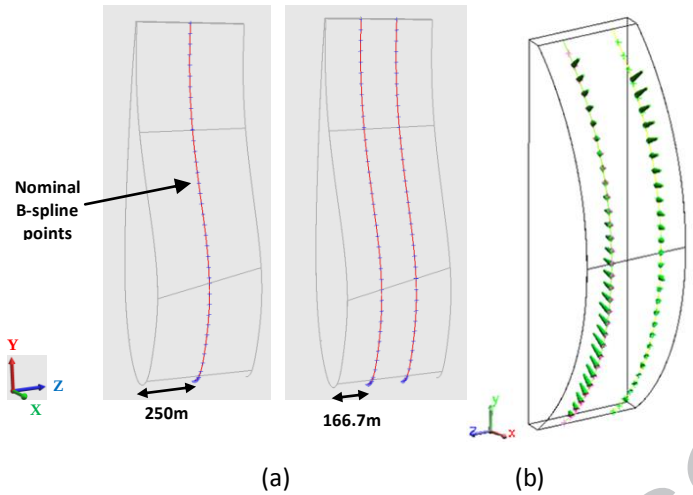


Figure 4. (a) Complex model, (b) simple D-shape model. Modified from [29,30].

In addition to the models, a full blade has also been constructed. This was loosely based on the Aerofoil schedule for the DOWEC 64.5m blade presented in [31]. Aerofoils used to construct the blade were taken from the UIUC Aerofoil Data Site [32], and can be seen in Figure 5.

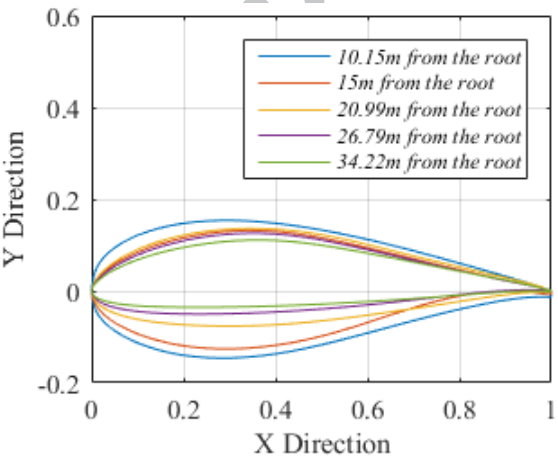


Figure 5. Aerofoil schedule for DOWEC 64.5m

*3.2 B-Spline implementation*

Experiments were first conducted on a variety of different square and cubic curves. This allowed the curves to be modified to any form before B-spline fitting was performed. Due to this, variations observed in the outputs were solely linked to the changes performed on the inputs. It also created the opportunity to edit a single feature within the fitting protocol at a time.

Matlab was used for B-spline curves and surfaces fitting. Although open source codes for B-spline fitting are available online, the code used within the current research was written by the author. This gave complete editorial freedom over the entire programme, thus allowing the implementation process to be examined and modified at any stage. This made it possible to investigate the areas of interest specific to this research, i.e. increase surface fitness via CP locations. Code was constructed for Equations 2, 3, 4, 5, 8, 9, 10 and 11. Figure 6 represents the basic algorithm used to conduct experimentation. During the analysis, the terms scaling, prominent direction, non-prominent direction and total angle turned have been used to make the analysis easier to grasp. Throughout this investigation these hold the meaning described below.

**Scaling** describes the amount a CP was increased or decreased by a percentage of itself. In mathematical terms it follows Equation 12

$$CP_{new} = CP_{old} + Scaling\ (CP_{old}) \qquad (12)$$

**Prominent direction** is defined by the direction in which the curve is travelling to reach its local minima or maxima.

**Non-prominent direction** is normal to that of the prominent direction.

**Total angle turned** refers to the angle made from a vector passing through the first and last measurement point and that of a line parallel to the prominent direction. The angle is positive if the direction from axis to vector is in the anti-clockwise direction.

After surfaces were fitted by the Matlab programme, curves could be fitted to the outputted results. For each fitted curve, an R squared value was generated. This ensured that the fitted curve conformed to the data. If the curve had an R squared value less than 0.8 the fit was scrapped and another was tried. This allowed a better understanding of the relationships to be found.

*3.3 Experiment protocol*

Throughout the research, the protocol laid out in Figure 6 was followed. This remained constant throughout the entire experiment with the exception of step 2. Once the optimal amount of blending had been found for each data set, it was kept constant. This ensured the greatest fit possible for each generated curve and surface.

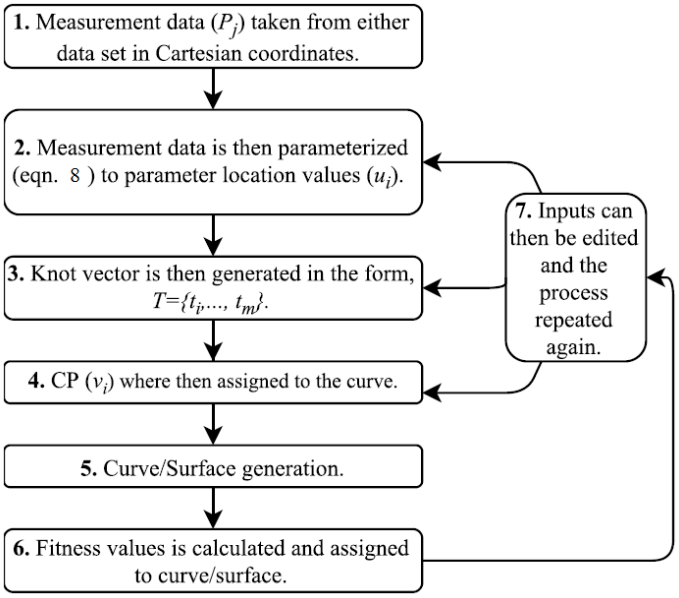


Figure 6. Experimental procedure used investigation and analyse

1. ***Measurement data ($P_j$) taken from either data set in Cartesian coordinates:*** A total of N measurement points were selected from any data set.
2. ***Measurement data ($P_j$) is then parameterized to parameter location values ($u_i$):*** The selected data set was then parameterized through Lees' blending equation presented as Equation 8 in the theory section.
3. ***Knot vector is then generated in the form, $T=\{t_i, ..., t_m\}$:*** An open uniform knot vector was selected. The total number of knots are found by solving Equation 7 for *m*
4. ***CP ($v_i$) were then assigned to the curve:*** CPs are selected in an array of manners which are discussed in the experiment section. These control points make up the control polygon of the spline.
5. ***Curve/Surface generation:*** With the outputs from steps 2-4, a curve or surface was generated through Equation 2 or 3.
6. ***Fitness values are calculated and assigned to curve/surface:*** Using Equation 10 the fitness value of that specific curve or surface can be calculated. This makes it possible to later justify the optimal CP location density and number.
7. ***Inputs can be edited and the process repeated again:*** Once the initial curve/surface had been generated, and the fitness value recorded, the inputs were changed and the algorithm was repeated.

The above protocol allows investigations into how the number of CP, the density of CP, the location of CP, the amount of blending, the data set used, and the degree affects the resultant fitness of the curve. By repeating the process, a different value for fitness is returned and comparisons can be made.

*3.4 Data analysis and ranking*

To justify the results obtained from the simulations, two ranking criteria were implemented. This made it possible to find the optimal combination of CP location, number and density that produced the most efficient curve/surface.

**Fitness:** Fitness was calculated using Equation 10. The closer the value is to one, the better the fit.

**Computational time:** As this study focuses its main effort on decreasing the through-put time of the QC process, it is important that a generated curve/surface not only has a high fitness value, but also a short computational time. As a result, each generation was timed and recorded where applicable.

*3.5 Experimental plan*

Throughout the following experiment the experiment protocol remained the same.

***Experiment 1 - Selection of parameterization method:*** In order to make sure that the correct parameterization method was selected, an algorithm containing Lees' blending equation was constructed. The algorithm varied the amount of blending in the range [0,1], in increments of 0.01. Each set of parameterized data was then constructed into a B-spline curve. From this, a value of fitness could be established, and the optimal amount of blending found. As measurements for both models were conducted differently, this experiment was undergone for both sets of measurement data as well for the DOWEC set.

***Experiment 2 - Effects of increasing the number of CP on computational time, and curve fitness:*** For the generated squared and cubic curves, CPs were evenly placed along the curve. A B-Spline curve could then be generated and the computations time along with the resultant fitness was recorded. The number of CPs was then increased and the process repeated.

***Experiment 3 – Effect of scaling CP location in both prominent and non-prominent directions:*** CPs were now scaled in the prominent direction by some magnitude away from

their original position in experiment 2. For each run a scale range was set. Each CP was scaled by the same scale. The algorithm returned a fitness for each different value of scaling. Taking the optimal amount of scaling in the prominent direction, CPs were then scaled in the other direction. Again a fitness value is obtained for each scale magnitude.

***Experiment 4 - Experiment 3 was repeated for squared and cubic curves that do not pass through $180^o$:*** By truncating the x data set, an angle could be obtained for the first and last data point. For this skewed curve, the location of CPs was varied in both directions. Again a fitness value was obtained for all values of scaling.

***Experiment 5 - Applying the information learnt, surfaces were constructed for all models:*** From all the previous experiments an initial CP location guide was created. This was then applied to the WT models. A fitness value was obtained from a surface with CP located on the data curve and then one from following the CP location guide.

## 4. Results and Analysis

### *4.1 Experiment 1- Selection of parameterization method*

To find the optimal amount of blending, *e* from Equation 8 was varied in the interval [0,1] in increments of 0.01. The fitness from the generated B-spline curve for each value of blending was calculated. The maximum fitness achieved for each data set and the corresponding amount of blending are presented in Table 2. For each data set the number of CP was kept constant and equal to 4.

| Data Set | Average Spacing Variation | Maximum Fitness | Optimal amount of blending (e) |
|---|---|---|---|
| Simple D Shape | 0.0408 | 3.79E-04 | 0 |
| Vestas Aerofoil | 0.2324 | 1.20E-05 | 0 |
| DOWEC Aerofoil | 0.3227 | 0.123 | 1 |

Table 2. Optimal amount of blending for each data set

The results in Table 2 show that for both the simple D shape and Vestas aerofoil the optimum parameterization method was the uniform method. Whilst using the DOWEC data set it was found that the optimum parameterization method was that of the cumulative chord length.

*4.2 Experiment 2- Effects of increasing the number of CP on computational time, and curve fitness*

While implementing the optimal parameterization method, CPs were placed evenly along both curves. This ensured that the number of CPs was the only influence on the resultant fitness. Figure 7 shows the relationship between the number of CPs and the obtained fitness.

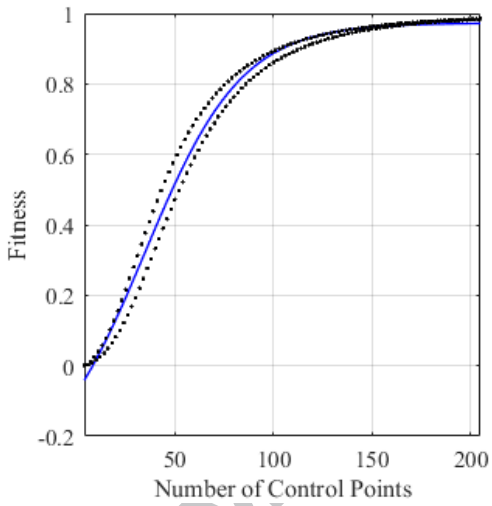


Figure 7. Fitness Vs number of CPs for squared

It was found that increasing the number of CPs along the squared curve caused the fitness to increase. Through curve fitting it was found that this held a sigmoid relationship. After a 125 CP the graph appears to plateau on fitness values of roughly 0.98. From Figure 7 it is interesting to note that there are two different sets of markers, one set above and the other below the best fit line. The experiment was then conducted for the cubic curve. The results of which are shown in Figure 8.

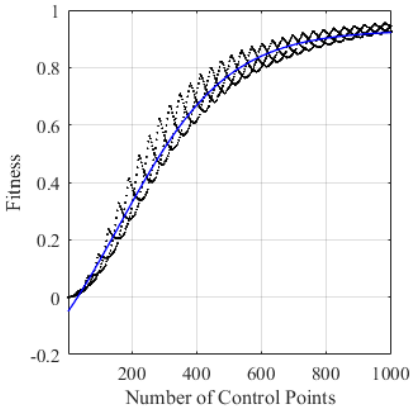


Figure 8. Fitness Vs number of CPs for cubic curve

Increasing the number of CPs for the cubic curve also produced a sigmoid relationship, with the fitness increasing. The cubic curves seems to have a maximum fitness of roughly 0.9. In addition, the maximum fitness is achieved with five times more CPs than the squared curve. This time it was not possible to group the data points into two clear sets.

As previously mentioned, the time to compute each B-Spline curve was recorded throughout. The time of computation vs the number of CPs for both curves can be seen in Figure 9.

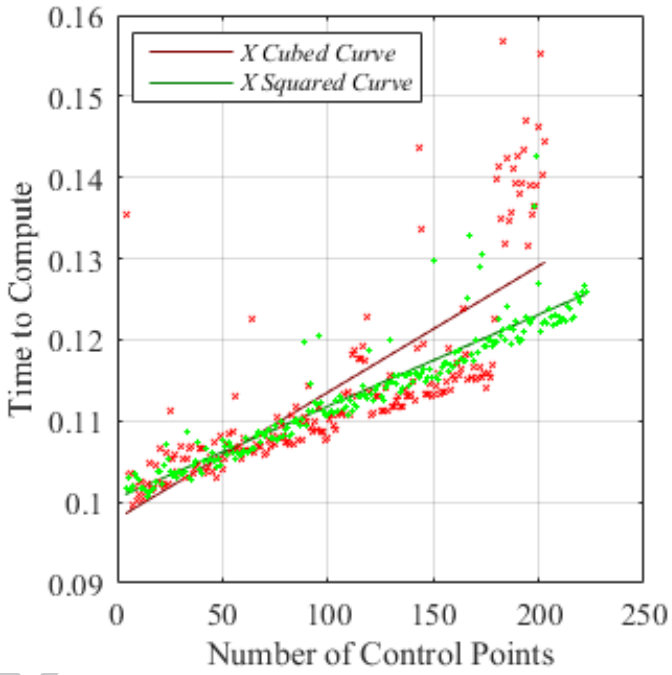


Figure 9. Computational Vs number of CPs

For both the squared and cubed curves the time vs number of CPs increased linearly up to the point when the number of control points reached 150. However, through the fitting process it was observed that the linear fit for the cubic curve was 27% better than that of the squared curve. In addition, it was found that the computational time for the cubed curve increased at a greater rate.

*4.3 Experiment 3- Effects of scaling CP location in both prominent and non-prominent directions*

To find the optimal CP location, the locations of the CPs were varied above and below its original location in the prominent direction. In the case of both the squared and cubic curves

this direction is denoted by the vertical line in Figure 10. Changes to the location were achieved by adding some fraction of the original CP value to itself. By using a scale spanning both negative and positive numbers, it was possible to find the optimal magnitude of scaling. The process can been seen in Figure 10.

Within Figure 10, the hashed grey lines show the control area enclosed by the CPs. The dotted black line depicts the ideal squared curve. The colour code refers to the fitness values obtained from each fitted B-spline curve. Scaling the CPs in the prominent direction caused the resultant B-spline curve fitness to increase until the maximum fitness was achieved, after which the fitness started to decrease. This is shown by the five coloured B-spline curves. Unsurprisingly, it was found that scaling the CPs in the opposite direction, to the arrow shown, yielded curves of a worse fit.

After the optimal amount of scaling in the prominent direction had been found and recorded, the CPs were scaled in the non-prominent direction. For the curve shown in Figure 10, this was scaling along a vector normal to the vertical black arrow. While scaling in this direction, the optimal amount of scaling in the prominent direction was kept constant. Curves with a total angle turned of zero achieved their maximum fitness with the CP located directly in line with the local maximum or minimum.

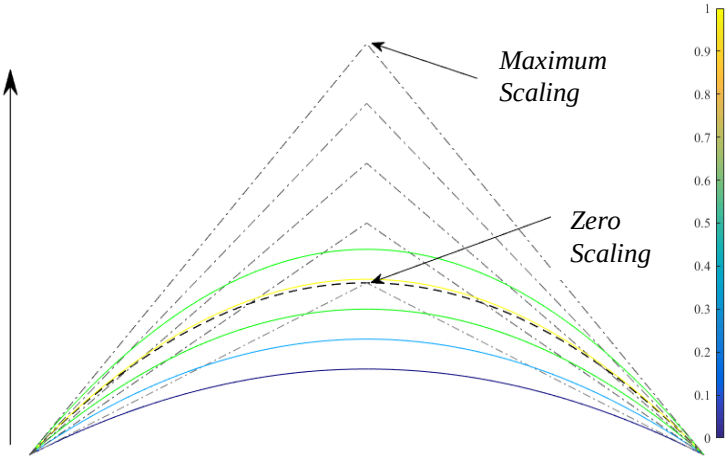


Figure 10. Effects of scaling CP location in prominent direction

*4.4 Experiment 4 – Repeat experiment 3 for squared and cubic curves that do not pass through 180 degrees*

To take this analysis further, the squared curves were modified in a variety of different ways. These included changes to the X and Y range of the curve, as well as changing the resultant angle between the first and last point. By doing so it was possible to create a chart that gave the ideal CP locations in relation to the local minima and maxima, for any curve. This was achieved by performing experiment 3 on the variety of modified curves. Firstly, the resultant angle was varied for different Y ranges. This can be seen in Figure 11.

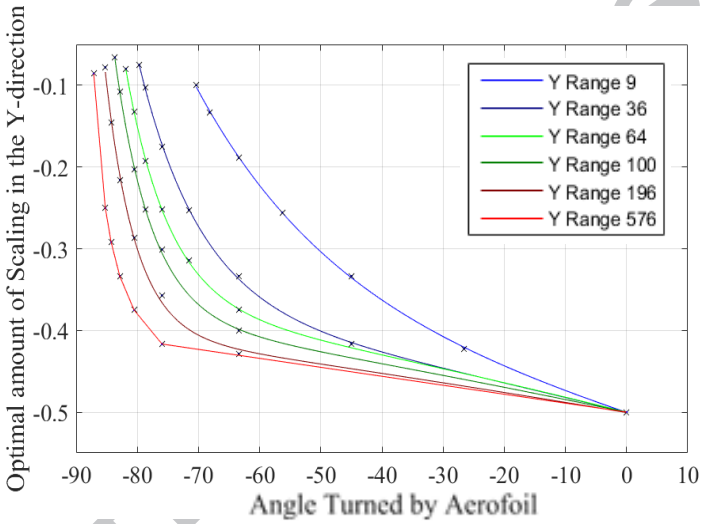


Figure 11. Optimal amount of scaling in the Y direction Vs angle turned by curve

Exponential curves were fitted to the data taken from the analysis. Figure 11 shows that as the total angle turned increases the amount of scaling in the Y needed to achieve an optimal fit decreases. Furthermore, as the Y range increases the average amount of scaling needed also increases.

Much like the protocol followed in experiment 3, once the optimal amount of scaling in the prominent direction had been found, it was kept constant. Scaling was then performed in the other directions. This again was carried out on all the modified curves used above. The results can be seen in Figure 12.

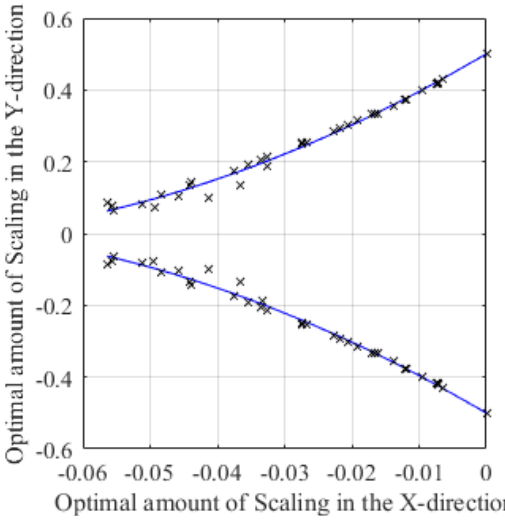


Figure 12. Optimal amount of scaling in the Y direction Vs optimal amount of scaling in the X direction

Each different Y range proved to have no effect on the optimal amount of scaling in the X direction. Furthermore, as the magnitude of scaling in the Y direction decreased the magnitude of scaling in the X increased. The two curves show that scaling should always be performed in the same directions that a vector would make if it was normal to a vector passing through the first and last data point.

*4.5 Experiment 5 – Applying the gathered information, surfaces were constructed for both models*

With both of the figures presented in the previous experiment, it is possible to find CP locations that would increase the B-spline's fitness. One simply needs to compute the angle between the first and last point of each curved section. For data sets of the DOWEC and Vestas aerofoils, this meant splitting up the aerofoil into subsequent sub-curves. Figure 13 shows how one of the Vestas aerofoils was split in order to find the angles. When dealing with the basic model, this was not necessary as the model only exhibited one curved section.

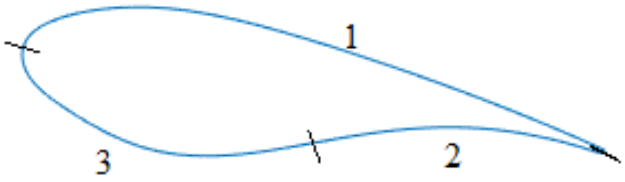


Figure 13. Splitting the Vestas aerofoil into curved segments

The aerofoil seen in Figure 13 was split into 3 sections, all resembling an x squared curve. For the bottom portion of the aerofoil, the split was made at the maximum gradient.

The first surface to fit with the CP location protocol, generated from experiment 3 and 4, was the D shape model. The results are shown in Figure 14, with the dashed lines representing the CPs.

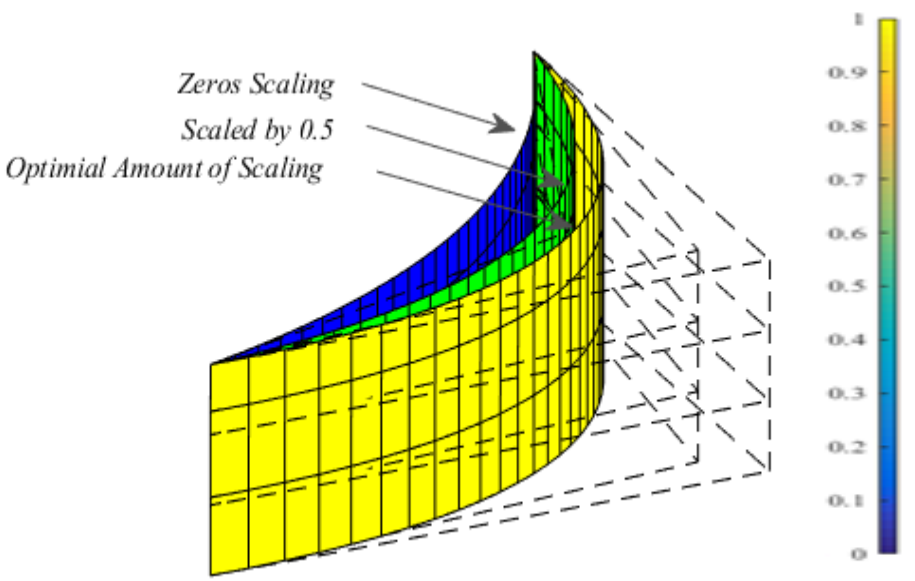


Figure 14. Applying CP location protocol to D shape aerofoil

The same colour scale as before was implemented to show how the fitness of the surface changed. The fitness values of each spline have been averaged to create a total fitness of the surface. These values are shown in Table 3.

| | Scaling Value | Surface Fitness |
|---|---|---|
| No Scaling | 0 | 0.00153 |
| Scaling Protocol | 0.5 | 0.00377 |
| Scaling from Optimiser | 0.77 | 0.0159 |

Table 3. Surface fitness for D shaped aerofoil

Although the fitness values achieved are relatively low, it has proved that by following the protocol the fitness value increased by roughly 150%. After passing the surface through an optimizer it was found that the ideal amount of scaling was 0.77.

In addition, the same experiment was conducted for the Vestas model. The generated surface can be seen in Figure 15 and Figure 16.

Figure 15 shows the blade from a side viewpoint. The two blades within the figure show one with and one without scaling. The fitness values for each amount of scaling are also presented in Table 4.

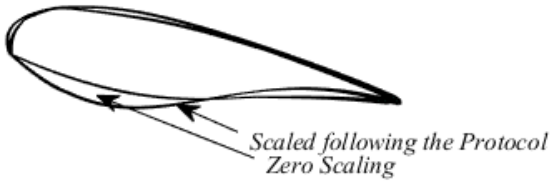


Figure 15. Complex aerofoil side

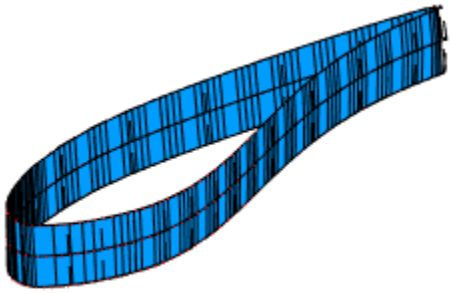

Figure 16. Applying CP location protocol to complex aerofoil

| Method | Surface Fitness |
|---|---|
| No Scaling | 0.000388 |
| Scaling Following Protocol | 0.0069 |

Table 4. Surface fitness for Vestas aerofoil

This is further verified by inspection of Figure 15. The shape of the generated surface with zero scaling is a long way off the ideal geometry. In comparison, the shape of the scaled surface conforms closely to the ideal shape, as seen in Figure 16 also.

Finally, the CP location protocol was applied to the full DOWEC blade model. The generated B-spline surface and the control volume are seen in Figure 17.

Again, in Figure 17 the dashed lines represent the control volume of the fitted B-spline surface. It was found that the fit for this data set preceded the fitness values of the two other models. Table 5 shows the fitness values for the both an unscaled and scaled model.

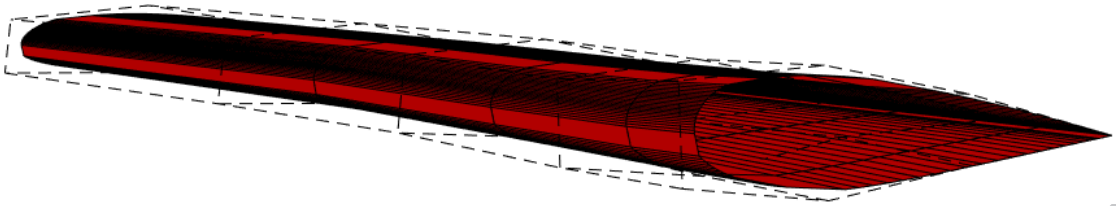

Figure 17. Applying CP location protocol to DOWEC

| Method | Surface Fitness |
|---|---|
| No Scaling | 0.0904 |
| Scaling Following Protocol | 0.1893 |

Table 5. Surface fitness for DOWEC aerofoil

The results from Table 5 show a 110% increase in the fitness from applying the CP location protocol. It should be noted that as this data set has a local maxima at the original of the working space, the aerofoil had to be shifted so that the centre became the original. This has to be done in order for the scaling to take any effect.

## 5. Discussion

### *5.1 Optimal parameterization method*

For the D shape model, as the spacing between data points was relatively uniform it was expected that maximum fitness would be achieved whilst using the uniform method. The DOWEC data set had large variations between consecutive data points. As a result, the chord length parameterization method gains maximum fitness as it takes into account the relative distances between the points.

From the spacing variation printed in Table 2, one would have also expected the same result for the Vestas aerofoil. Examining the data set closer found that there were three distinct sections. Within the three sub-data sets, variations were in the order of E-15. As a result, the resultant curve can be thought of as three sections of uniformly spaced points, leading to the uniform blending method.

*5.2 Effects of increasing CP on computational time and fitness*

For the squared curve, it was found that increasing the number of CP from 10 to 100 had the greatest effect on the fitness. Before and after that range the effects were less prominent. In addition, it was found that an even number of CPs always generated a curve to higher fitness values. This can be seen in Figure 7 with the even number of CP as markers above the best fit line.

Similar results were found for the cubic curve. This time the range of maximum effect was between 50 and 600 CP. However, this time it was not possible to quantify whether an odd or even number of CP resulted in a higher fitness. Instead, it was observed that CP, which were multiples of three, generated higher fitness values, seen by the braided pattern of the markers in Figure 8. It was found that the squared curve obtained a larger maximum fitness.

What is interesting to note is that, in a sub-investigation, implementing this experiment to the simple D shape and Vestas data set yields similar results. However, once the number of CP exceeded the number of data points there was a sudden drop in fitness, after which the fitness carried on increasing.

*5.3 Effects of scaling in both the prominent and non-prominent directions*

Moving the CP location away from the ideal curve caused the fitness to increase. For curves with a total angle turned equal to zero, the optimal amount of scaling was equal to half. Whilst using the cubic curve it was found that having a CP at maximum gradient caused the overall fitness to decrease, and caused the computational time to increase.

*5.4 Effects of using a curve that does not pass through 180°*

Although Figure 11 only shows an angle range of -90° to 0°, analysis was also carried out over the range [-90°:90°]. The same magnitude of scaling was achieved for range [-90°:0°] and [0°:90°], but in the second range the amount of scaling was positive. On top of this it was found that scale should be performed in the same directions as a normal to the vector passing through the first and last data point.

In order to create the CP location protocol, the results had to be normalised. This was done by dividing each scaling value by its prospective axial range. Figures 11 and 12 show the

normalised results. By normalizing the data it was possible to use the guide for any curve. To use the guide an operator needs to simply find the angle turned by each curved section of the blade, along with the prominent axial range of that section. It is then possible to go to Figure 11 and retrieve the optimal amount of scaling in the Y direction, after which Figure 12 is used to relate this to the optimal amount of scaling in the X direction.

*5.5 Appling the CP location protocol*

Applying the CP location protocol to the three data sets increases the fitness values. In the case of the D shape aerofoil it was shown that the protocol provided scaling that was only 17% off the optimal amount of scaling. It is also important to note that in all three models the number of CPs used was kept to a minimum. Following the protocol, therefore, increases the fitness whilst decreasing the number of CPs and this decreases the amount of time needed to compute the surfaces. Results can be seen in Figure 18 and Figure 19.

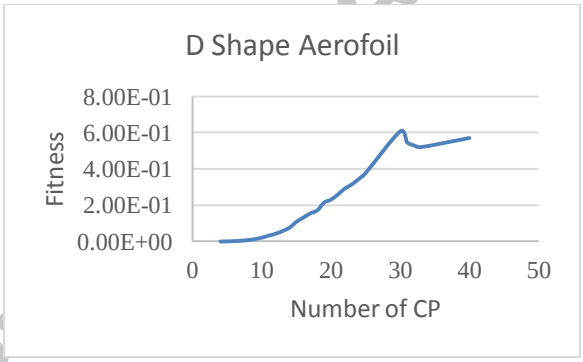


Figure 18 Number of control points Vs Fitness for D shape aerofoil

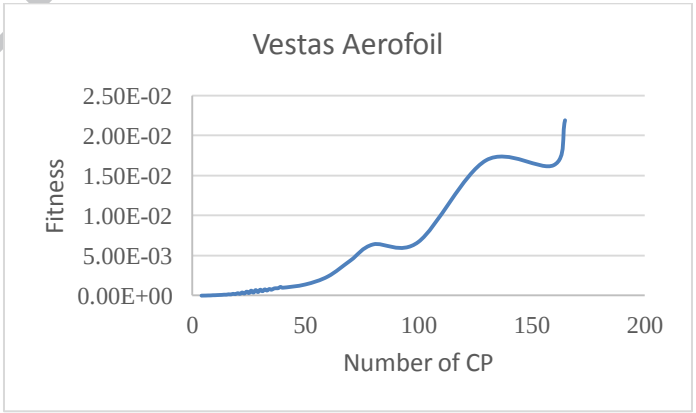


Figure 19 Number of control points Vs Fitness for Vestas aerofoil

## 6. EXPERIMENTAL EVALUATION AND RECOMMENDATIONS

The CP location protocol relies on information gathered from a large amount of experiments. AlthouMgh the protocol has generated curves and a surface with promising fitness values, performing more experiments will increase the accuracy of the protocol. Undergoing similar analysis on a greater variety of different shaped curves will not only increase the protocol capability but also decrease its uncertainty. Due to this, it is recommended that this experiment protocol is carried out on a greater variety of curves.

During the research, the focus of analysis has been on the QI stage of WT blades. With such products, the tolerances are high and inspection rate low. As a result, the protocol has to be highly accurate. The protocol should be applied to a product of higher production with lower tolerance. By doing so, the need for optimizers in B-spline surfaces could be removed altogether if high fitness values are seen.

## 7. CONCLUSION

This investigation has found that by following a simple CP location protocol, the original fitness of a B-spline surface or curve can be increased by up to 150%. Additionally, placing CP above/below local maxima and minima respectively yielded optimal fitting results for curves that have a resultant angle equal to zero.

Furthermore, scaling should always be performed in the same direction that a vector would make if it was normal to a line passing through the first and last data point of a particular curved section. A simple protocol for CP location has also been constructed in the form of two charts. This allows CPs to be placed in locations that will ensure a good fit first time round. If tolerances of products are high this also means that later optimization computation times will be reduced.

## ACKNOWLEDGMENT

This research is part of the wind turbine structural condition monitoring program funded by EPSRC through the Centre for Through Life Engineering Services (EP/1033246/1, Project SC006) and EPSRC Impact Acceleration Award (EP/K503368/1) and Royal Society – Natural Science Foundation China International Exchange award (IE150600).

## REFERENCES


[1] Department of Energy and Climate Change, "UK Renewable Energy Roadmap Update 2013", London, 2013.

[2] Department of Energy and Climate Change, "Renewable sources of energy: Chapter 6, Digest of United Kingdom Energy Statistics (DUKES)", London, 2015.

[3] D. Cairns, J. Nelson and T. Riddle, "Wind Turbine Composite Blade Manufacturing: The Need for Understanding Defect Origins, Prevalence, Implications and Reliability", Sandia National Laboratories, 2011.

[4] European Wind Energy Association, "The Economics of Wind Energy", Denmark, 2009.

[5] "Suzlon Blade Recall: Retrofit Program to Address Cracking: CompositesWorld", Compositesworld.com, 2016. [Online]. Available: http://www.compositesworld.com/news/suzlon-blade-recall-retrofit-program-to-address-cracking.

[6] Y. Gong, R. Johnston, C. Melville and E. Seibel, "Axial-Stereo 3-D Optical Metrology for Inner Profile of Pipes Using a Scanning Laser Endoscope", International Journal of Optomechatronics, vol. 9, no. 3, pp. 238-247, 2015.

[7] M. Rak, A. Wozniak and J. Mayer, "The use of low density high accuracy (LDHA) data for correction of high density low accuracy (HDLA) point cloud", Optics and Lasers in Engineering, vol. 81, pp. 140-150, 2016.

[8] Q. Wang, N. Zissler and R. Holden, "Evaluate error sources and uncertainty in large scale measurement systems", Robotics and Computer-Integrated Manufacturing, vol. 29, no. 1, pp. 1-11, 2013.

[9] B. Gapinski, M. Wieczorowski, L. Marciniak-Podsadna, B. Dybala and G. Ziolkowski, "Comparison of Different Method of Measurement Geometry Using CMM, Optical Scanner and Computed Tomography 3D", Procedia Engineering, vol. 69, pp. 255-262, 2014.

[10] F. Franceschini, M. Galetto, D. Maisano and L. Mastrogiacomo, "Large-scale dimensional metrology (LSDM): from tapes and theodolites to multi-sensor systems", International Journal of Precision Engineering and Manufacturing, vol. 15, no. 8, pp. 1739-1758, 2014.

[11] W. Cuypers, N. Van Gestel, A. Voet, J. Kruth, J. Mingneau and P. Bleys, "Optical measurement techniques for mobile and large-scale dimensional metrology", Optics and Lasers in Engineering, vol. 47, no. 3-4, pp. 292-300, 2009.

[12] J. Nelson, D. Cairns and T. Riddle, "Manufacturing Defects Common to Composite Wind Turbine Blades: Effects of Defects", Montana State University, Bozeman, 2011.

[13] N. Judd and W. Wright, "Voids and their effects on the mechanical properties of composites- an appraisal", Sample Journal, vol. 14, no. 10-14, 1978.

[14] H. Haron, A. Rehman, D. Adi, S. Lim and T. Saba, "Parameterization Method on B-Spline Curve", Mathematical Problems in Engineering, vol. 2012, pp. 1-22, 2012.

[15] I. Schoenberg and C. Boor, I.J.Schoenberg selected papers. BirkhÃuser Boston, 1945, pp. Vol. 2, 45-46.

[16] C. de Boor, "On calculating with B-splines", Journal of Approximation Theory, vol. 6, no. 1, pp. 50-62, 1972.

[17] P. Eilers and B. Marx, "Flexible smoothing with B -splines and penalties", Statistical Science, vol. 11, no. 2, pp. 89-121, 1996.

[18] R. Farouki and J. Hinds, "A Hierarchy of Geometric Forms", IEEE Comput. Grap. Appl., vol. 5, no. 5, pp. 51-78, 1985.

[19] E. Lee, "Choosing nodes in parametric curve interpolation", Computer-Aided Design, vol. 21, no. 6, pp. 363-370, 1989.

[20] Y. Sun, J. Weil and D. Xia, "Parameter optimization for B-spline curve fitting based on adaptive genetic algorithm", Journal of Computer Applications, vol. 30, no. 7, pp. 1878-1882, 2010.

[21] M. Floater, "Parameterization", Computer Science Department, Oslo University, 2011.

[22] W. Ma and J. Kruth, "Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces", Computer-Aided Design, vol. 27, no. 9, pp. 663-675, 1995.

[23] V. T. Dung and T. Tjahjowidodo, “A direct method to solve optimal knots of B-spline curves: An application for non-uniform B-spline curves fitting”, PLoS One, Vol. 12, No. 3, e0173857, 2017.

[24] H. Park, “B-spline surface fitting based on adaptive knot placement using dominant columns”, Computer-Aided Design, Vol. 43, pp258-264, 2011.

[25] Y. H. Zhang, J. Cao, Z. G. Chen, X. Li, X. M. Zeng, “B-spline surface fitting with knot position optimisation”, Computer & Graphics, Vol. 58, pp73-83, 2016.

[26] W. N. Zheng, P. B. Bo, Y. Liu, W. P. Wang, “Fast B-spline curve fitting by L-BFGS”, Computer Aided Geometric Design, Vol. 29, No. 7, pp.448-462, 2012.

[27] W. P. Wang, H. Pottmann, Y. Liu, “Fitting B-spline curves to point clouds by curvature-based squared distance minimization”, ACM Transactions on Graphics, Vol. 25, No. 2 pp214-238, 2006.

[28] P. B. Bo, G. N. Luo & K. Q. Wang, “A graph-based method for fitting planar B-spline curves with intersections”, Journal of Computational Design and Engineering Vol. 3, pp14-23, 2016.

[29] J. Talbot, Q. Wang, N. Brady and R. Holden, "Offshore wind turbine blades measurement using Coherent Laser Radar", Measurement, vol. 79, pp. 53-65, 2016.

[30] A. Summers, (April 2013), “Investigating the Measurement of Offshore Wind Turbine Blades Using Coherent Laser Radar”, MEng Technical Report, Durham University.

[31] Sandia National Laboratories, "Definition of a 5MW/61.5m Wind Turbine Blade Reference Model", Albuquerque, 2013.

[32] "UIUC Aerofoil Data Site", M-selig.ae.illinois.edu, 2016. [Online]. Available: http://m-selig.ae.illinois.edu/ads/coord_database.html. [Accessed: 23- Apr- 2016].

**Research Highlights**

- This paper details a quality inspection process for offshore wind turbine blades.
- Different curve fitting methods were investigated.
- 
- Analysis of the relationship between b-spline curve generation and control points has been carried out.
- Quantification of scaling in both the prominent and non-prominent directions was discussed.